# Team Orange’s Cross-Topix

Bridging Music Silos using Semantic Technologies

X-Informatics Final Project, Spring 2011

Professor Peter Fox

Tetherless World Constellation

Rensselaer Polytechnic Institute



# Team Members:

**Colin Anderson** - Second semester MS in IT, MIS concentration  
**Yu Chen** - 1st Year PhD ECSE, Tetherless World Constellation 518-522-7669  
**Samuel Johnson** - 4th semester (part time) MS IT, HCI Concentration  
**Tim Lebo** - 2nd Year Ph.D. Cognitive Science, Tetherless World Constellation  
**Amanda Olyha** - Senior, Physics and CS

Contents

[Team Orange’s Cross-Topix 1](#_Toc291793204)

[Team Members: 1](#_Toc291793205)

[Introduction: 3](#_Toc291793206)

[1. Use Case Development: 3](#_Toc291793207)

[“See Also” Box: 3](#_Toc291793208)

[Social Machine: 6](#_Toc291793209)

[Prototype Design: 8](#_Toc291793210)

[Information Uncertainty: 8](#_Toc291793211)

[Semiotics: 8](#_Toc291793212)

[Cognition: 8](#_Toc291793213)

[Architecture: 8](#_Toc291793214)

[2. 9](#_Toc291793215)

[Concept Model: 9](#_Toc291793216)

[Logical Model: 10](#_Toc291793217)

[3. 10](#_Toc291793218)

[Appendix 10](#_Toc291793219)

[A. Information Uncertainty Clarification: 10](#_Toc291793220)

[B. Scrape Specifics: 13](#_Toc291793221)

[C. Crawl and Turtle Documentation: 14](#_Toc291793222)

[D. Apply String Matching Algorithm 15](#_Toc291793223)

[E. Implementation of the Social Machine: 17](#_Toc291793224)

[F. Implementation of the “See Also” box: 20](#_Toc291793225)

[G. Install Triple Store and SPARQL Endpoint 22](#_Toc291793226)

[H. Future improvements to the system 26](#_Toc291793227)

[I. Project Pre-Definition: 27](#_Toc291793228)

[J. Proposals 28](#_Toc291793229)

[K. Project Definition and Decomposition 32](#_Toc291793230)

[L. Meeting Notes 34](#_Toc291793231)

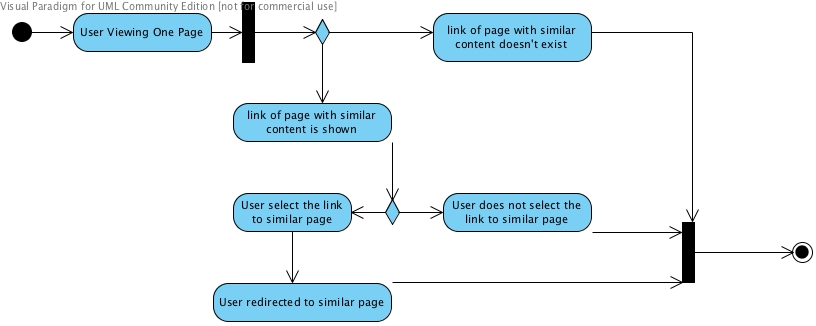
# Introduction:

Choosing public domain music scores that are appropriate for instruction and performance can be challenging.. For example, users may want a a high quality score in a certain genre or by a specific composer, but it is often difficult to find these scores in the existing systems. Therefore, our group chose to focus on developing a system dealing with library informatics. Our group wanted to use a similarity heuristic combined with crowd sourcing to associate webpages on two separate sites. Through the development of a “Social Machine” and a “See Also” box, musicians are able to find similar pages on different sites based on multiple parameters. It adds value to the user by allowing them to easily find the same music on different sites and it also improves both sites by creating greater resource acquisition.

# 1. Use Case Development:

## “See Also” Box:

**What is it?  Why did we make our choice?**  
There are two main music score wikis that are available to the public: the [Petrucci Music Library](http://imslp.org/wiki/) and the [Choral Public Domain Library](http://www3.cpdl.org/wiki/).  Both of these libraries have thousands of pages for individual music scores and composers.  In many cases, the same pieces of music and the same composers can be found on both sites, but different information may be found on either site.  We thought it would be helpful if we could provide a way for users to know that there exists this other site that contains the same information, but may be in a different format or a different edition of the publication.  We decided that the best way to convey this information would be as a “See Also” box that would give suggestions to point the user towards pages or categories in the other wiki.  This may expose the reader to new music or provide another source of the sheet music that may be in a more convenient format.  
  
**Use Case Name**  
Webpage Congruency Identifier Box  
  
**Goal**  
The users of the [Petrucci Music Library](http://imslp.org/wiki/) and/or the [Choral Public Domain Library](http://www3.cpdl.org/wiki/) will have a way to connect the information on both sites.  This will not make use of a third party site, but will embed the information onto the current sites to provide a direct link between the two services.  
  
**Summary**  
The user would access the Petrucci Music Library and/or the Choral Public Domain Library to find music scores based on the search categories of composer, title, or date of publication, among others. Lacking viable information on one site, the user would want to find another site to satisfy their needs. For instance, the Petrucci Music Library has hand written scores which are difficult to read. Therefore, moving to the Choral Public Domain Library one might find one that is typeset. The cross-topix infrastructure is embedded on both sites to allow users to access the other site’s similar listings.

**Activity Diagram**

**Actors**  
Primary: Teachers  
Primary: Musicians  
  
These actors initiate the presentation of the webpage congruency identifier box after searching or browsing a page on either Wiki site.  
  
**Preconditions**

* There is at least one page on the wiki that the user can view
* The sites are up and available
* The plug-in is functioning
* The user has Internet access  
    
  **Triggers**
* The user views a page on the wiki for a specific music score
* The user views a page on the wiki for a specific composer

**Basic Flow**

* The user views a page on the first wiki.
* A box is displayed on the page that contains links to the same information on the second wiki.
* The user selects one of the links
* The user is redirected to the selected page on the second wiki

**Alternate Flow**

* If there is not a page on the second wiki that can be linked, a box will still be displayed, but it will not contain links to pages on the second wiki.
* If the user does not select one of the links (User does not complete step 3), the user will remain on the first wiki and will not be redirected to the second wiki.

**Post Conditions**  
The user is viewing either a composer page or a music score page on the second wiki.  
  
**Resources**

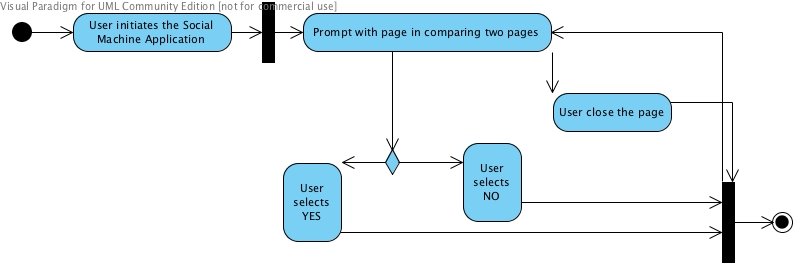
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Data** | **Type** | **Characteristics** | **Description** | **Owner** | **Source System** |
| Petrucci Music Library | Remote |  | Wiki music information site | Project Petrucci LLC | MediaWiki |
| Choral Public Domain Library | Remote |  | Wiki music information site |  | MediaWiki |

**Link:**

http://leo.tw.rpi.edu:81/cross-topix/see-also-box/hello.php

## Social Machine:

**What is it?  Why did we make our choice?**  
In order to for the “See Also” to work, we created a heuristic that performed string comparisons on the titles of the pages in the two sites. We did not believe this was adequate for the system to function at the level of accuracy that we would like, so we created a “Social Machine” that allowed users to compare the system generated comparisons and judge their validity.  
  
**Use Case Name**  
Social Machine: Crowd source information accuracy relational application  
  
**Goal**  
The users of the Petrucci Music Library and/or the Choral Public Domain Library will have a way to communally develop associations between the pages of composers and scores.    
  
**Summary**  
The user would access the Petrucci Music Library and/or the Choral Public Domain to help create associations between the sites based on composer and score. The user would go to the “Social Machine” section. The user would then be prompted to decide whether the two links were associated through a yes or no response. This activity is designed to give users the ability to cross reference and to find the information on both sites. This allows the user input to add another layer of sophistication to creating associations. Prior to this application, string matching was the only way to cross reference the material. Through crowd sourcing it enables better accuracy of associations.

Activity Diagram  


**Actors**  
Primary: Teachers  
Primary: Musicians  
Primary: Anyone involved in the community  
  
These actors repeatedly answer association questions until they no longer choose to do so.

**Preconditions**

* There are pages on the wikis to compare
* The sites are up and available
* “Social Machine” is functional
* The user has Internet access

**Triggers**

The user initiates the “Social Machine” application

**Basic Flow**

* The user initiates the “Social Machine”
* The user is prompted to decide whether two web pages from the separate wikis are associated with a yes or no response. The decision is based on attributes pulled from both the wikis and links to the pages for further information gathering.
* After the choice is made, another combination is displayed.
* The process is iterated until the user ceases to partake in the application.

**Alternate Flow**  
There are none.

**Post Conditions**  
The user is viewing either a composer page or a music score page on the second wiki.  
  
**Resources**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Data** | **Type** | **Characteristics** | **Description** | **Owner** | **Source System** |
| Petrucci Music Library | Remote |  | Wiki music information site | Project Petrucci LLC | MediaWiki |
| Choral Public Domain Library | Remote |  | Wiki music information site |  | MediaWiki |
| Association Data Store | In  situ | Unmanageable | List of all possible combinations of sites between the two wikis | Orange Team | RPI Server |

**Link:**

http://leo.tw.rpi.edu:81/cross-topix/socialMachine/helloWorld.html

### Prototype Design:

The prototype will connect the same pages on the Petrucci Music Library and/or the Choral Public Domain sites. When the user searches for music based on composer, genre, era, etc. they will eventually come to the page that they desire. The “See Also” box will be displayed and let the user know what pages on the other site coincides with the page currently viewed. In addition, the user can also help validate comparisons using the “Social Machine.” Through this, they will be able to rapidly go through selections and help with making decisions about correctness of the heuristic test. There will be an ability to look at each site to see if the pages coincide.

### Information Uncertainty:

Information uncertainty is one of the greatest problems in the system. The ETL sequence is sound using the variety of scripts and SPARQL query end points. After the ETL, the system uses Levenshtein’s distance algorithm which increases precision by eliminating a large portion of invalid combinations. Despite the increased recall and greatly reduced fall-out, it does not give complete certainty that the matches are correct. Therefore, to reach the level of sufficient confidence, we added the “Social Machine” to include human decision and decrease the overall entropy of the system. However, these results are only validated if the human interaction with the machine is ethical.

The information uncertainty of the “See Also” box is directly related to the information certainty created by the “Social Machine.” Therefore, based on the level of sufficient completeness provided by the “Social Machine,” one can decipher the validity of the “See Also” box.

See Appendix A for thorough reflection on information uncertainty process.

### Semiotics:

The system has a visual scoring system based on the results from the machine heuristic, which is determined bye Levenshtein’s algorithm. This algorithm gives a pair of titles a score on the range of 0 to 1 with 0 as a “perfect match”. Our scoring system is a three star system that has been vetted by the team’s sample segment to represent the accuracy of the combinations. Three-stars represents a value between 0 and 0.33. This is considered a probable perfect match. A two-star rating would range from 0.34 to 0.64. These represent a good probability of matching. A score between 0.65 and 0.8 is considered one-star. It is unlikely there is a match but there is a slim chance. From 0.8 to 1 statistically, there is no chance of a match. We chose the three-star rating because it is common and easily interpreted.

### Cognition:

The team used library informatics as the basis of our system. The team took collections of music that were cataloged and classified on two separate Wikis. They have been organized in multiple ways including composer, era, genre, etc. Access to these sites is unlimited. With all this music in the public domain there are no proprietary, legal, or publishing issues. Because of the organization of the material, the team attempted assist with cognition by allowing a user’s expertise to enhance the validity of the system by adding a “Social Machine”. At the same time, the system eliminates issues of uncertainty for both experts and novices by deciphering commonalities between the related Wiki pages with the “See Also” box. Finally, we attempted to create a system that was intuitive to the nature of the user’s search by attempting to match pages on the separate sites that they would need.

### Architecture:

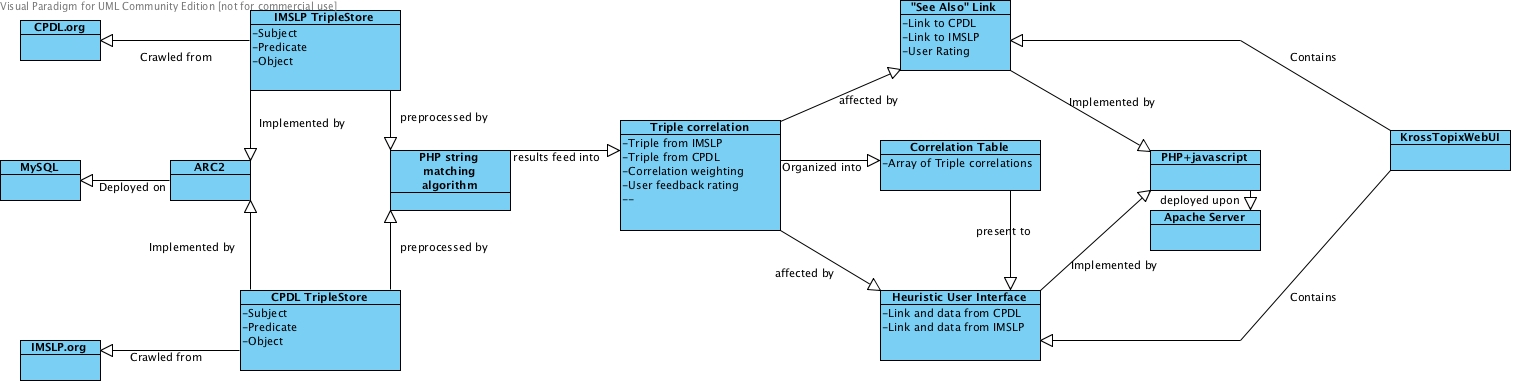
In acknowledging the distributed nature of content creation and consumption, interaction with our prototype uses the foundations of the web architecture in our information modeling and service invocations.

# 2. Models

## https://lh3.googleusercontent.com/ZOBqYP2lT7B7kWhw_vtWKH-NEBOgU6hZixxQMwpStDH41J_40hBA--E5K9Mzc1IJDmb-7uInftrlJ42i_-nCTPA-XHchTf5oIgxANwoOoJBuFZUkSikConcept Model:

The above diagram is the concept model of our system. The designed model begins with the web pages on the CDPL.org and the IMSPL.com sites. The URLs and titles are scraped (See Appendix B for specificsand then encoded in RDF using Turtle (See Appendix C for specifics). The results are stored in a SPARQL endpoint for later use. The information is then run through a string matching algorithm to test their similarity and reduce uncertainty (See Appendix D for specifics). After the string matching algorithm is applied, the correlations are stored on a named graph and are presented to the “Social Machine” and the “See Also” interface (See Appendices E and F for specifics). At the social machine, the user views the other site and determines if the connection is valid. The results are fed back to the correlation table stored in the triple store database. The information throughout the system is provisioned to allow the user to infer the greatest amount of probable information. This allows the user to focus on information uncertainty of a minor combination of pages instead of over half a billion possible matches.

## Logical Model:



The above diagram is the logical model for the system. First, the raw data was scraped from the two wikis. That raw data is organized in the triple store subject, predicate, and object tuples and then stored in ARC2. All the tuples in the triple store are processed with the string matching algorithm. The algorithm evaluates the similarities between the two web pages and helps to reduce uncertainty. The results of the processing algorithm are fed into a named graph on a SPARQL endpoint. This information is available to anyone inside the RPI system (See Appendices G for details). Later the data in the named graph can be modified by the user heuristic operation. The results of the user heuristic operation will be resent to the named graph endpoint, which will help to modify the correlation between the two webpages. All those results will be presented to the PHP based web user interface. Each of the heuristics reduces uncertainty, while provisioning of information is executed throughout the system as information moved from the original websites to the display of information to the user.

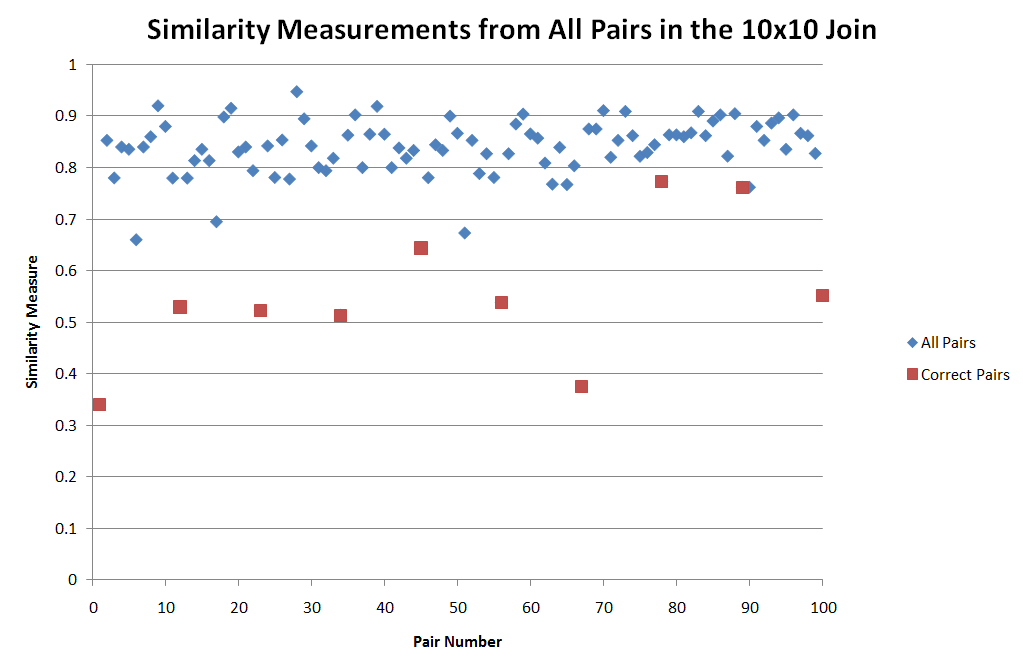
# 3. Prototype Implementation

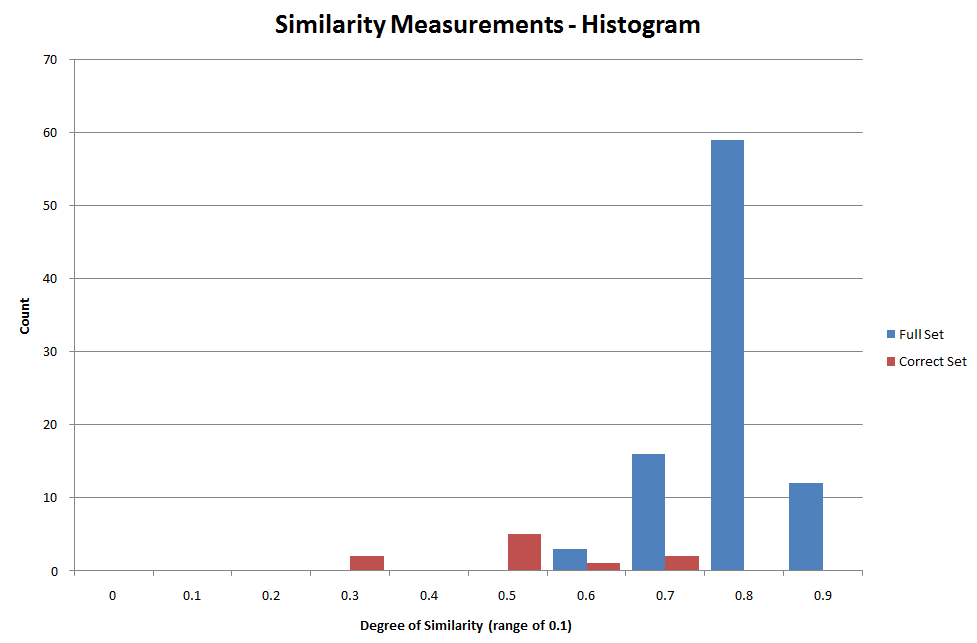
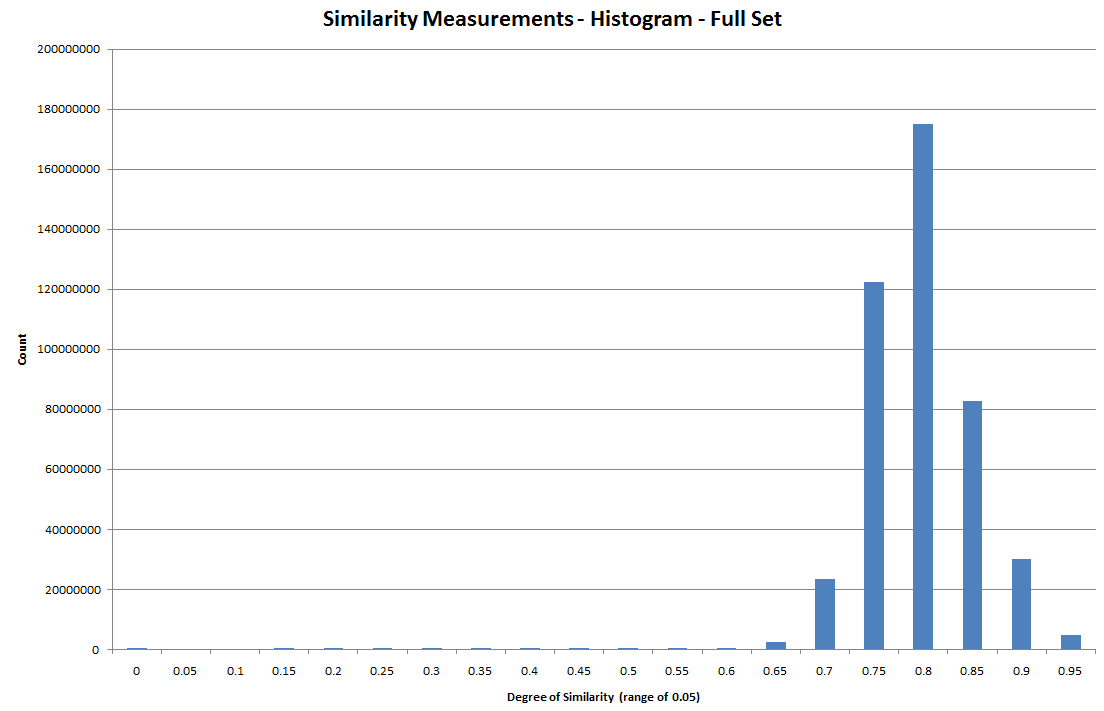
The team developed their system based on the existing architectures, storage of the gleaned information, dissemination of the information, and representation of the information. The first decision was how to glean the information for the two web sites. The team designed a scrape function to get the URLs and titles (See Appendix B). That data was stored in RDF using Turtle and placed in a triple store using an ARC2 interface to a MySQL database (See appendix G). The team then chose Levenshtein’s distance algorithm to evaluate the similarities between the sites. The team chose the algorithm for its common use and its reliability. This method is not very fast and because of this the team decided to store the information on a named graph on a SPARQL endpoint. This would eliminate the need to reprocess the information each time the system ran. The drawback is that it needs periodic updates (See G for details) since users can add new music to the sites which will not be known to our endpoint. Using the machine heuristic alone did not offer enough validity to the system, so the team chose to add a human heuristic as well to further reduce uncertainty in the system. The team decided to use JavaScript for the “Social Machine” and PHP for the “See Also” box. The Wikimedia is PHP based and used on both websites, so insertion of this code will be simple. The “Social Machine”, on the other hand, is client heavy so one needs Ajax for that configuration (See Appendices E and F). The “Social Machine” and “See Also” boxes were designed to take the user into account. The functioning of the system is invisible to the user and a three-star rating system was implemented to enhance understanding of the matching.

# Appendix

## A. Information Uncertainty Clarification:

In order to establish a ground truth Sam and Amanda hand-selected links and test cases between the two wikis, this provided 10 test cases. The team then ran the hand-picked test cases through the string joiner. The results were satisfactory, although there will be special cases where two terms may have the same meaning but completely different names.

**Histograms**  
  
Figure: Values of the similarity measures for all pairs created by the joining of the 10-count sets.  The x axis is irrelevant - it is the “pair number” (so the first pair to be compared is 1, the second pair is 2, etc.).  The y axis is the actual value returned by the function.  See the chart below for a histogram broken down into 0.1 increments.  
  
  
These values were calculated by comparing 10 pages from IMPSLP.org with 10 pages from CPDL.org. This gives us a total of 100 possible pairs with only 10 of those truly correct.  These 10 correct ones are distinguished by the red points.  According to this algorithm, a smaller similarity value means that the two titles are more similar (they have a smaller difference).  As seen in the above graph, the red points are (on average) much lower than the blue points which gives us the impression that the values returned by our similarity measure are consistent with the actually cases.    
  
A histogram of the above points was created with bins of size 0.1.  It can be found below.  We see that the majority of the full sets of pairs are found in the 0.7-1.0 range while our correct pairings are found on the 0.3-0.7 range.  There is some overlap, but it is very small when compared to the overall picture.

  
Figure: Histogram of the similarity measurements from the 10 x 10 hand-picked examples.  The red bars are the correct matchings, while the blue bars are the all the others as a result of the join.  The values are broken into 0.1 value increments, with the displayed number being the lower bound.  
  
To see how our similarity matching algorithm performs on a larger data set, we ran it on the full listings of both wikis.  We had a total of 441,736,711 pairs and the resulting histogram can be found below.  We found that over 99% of the comparisons had a similarity measurement greater than 0.7.  Only 3,331,078 (0.75%) had values less than 0.7.  If we restrict our dataset to only pairs with similarity values less than 0.65, we bring our data set to 410,272 pairs, which is just under 0.1% of the total.  We used this information to construct our star rating system for use with the “See Also” box.  
  
  
  
Figure:  Histogram as a result of the full join.  Bins are of size 0.05 in this graph.   The displayed value is the lower bound.

## 

## B. Scrape Specifics:



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Scrape  Starting Point | Scrape Results | **Source** Identifier | **Dataset** Identifier | **Version** Identifier | Dataset URI |
| [Special:AllPages](http://imslp.org/wiki/Special:AllPages) | [imslp.ttl](https://github.com/samuelbjohnson/cross-topix/blob/master/page-titles/imslp.org/manual/index-pages/index-pages/aggregate/imslp.ttl) | imslp-org | wiki-pages | 2011-Apr-14 | [2011-Apr-14](http://leo.tw.rpi.edu/source/imslp-org/dataset/wiki-pages/version/2011-Apr-14) |
| [Special:AllPages](http://www3.cpdl.org/wiki/index.php/Special:AllPages) | [cpdl.ttl](https://github.com/samuelbjohnson/cross-topix/blob/master/page-titles/cpdl.org/manual/index-pages/aggregate/cpdl.ttl) | cpdl-org | wiki-pages | 2011-Apr-14 | [2011-Apr-14](http://leo.tw.rpi.edu/source/cpdl-org/dataset/wiki-pages/version/2011-Apr-14) |

## 

## C. Crawl and Turtle Documentation:

(X) Design Crawl encoding using RDF. (Tim)

<http://dublincore.org/documents/dcmi-terms/#terms-title>

(X) Encode scrape results using the Turtle syntax. (Tim)  
  
The following [Turtle syntax](http://www.w3.org/TeamSubmission/turtle/) for our RDF [abstract model](http://www.w3.org/TR/rdf-concepts/) shows an example of the data provided by the scraper. The *subjects* of the triples are the pages themselves, the [dcterms:title](http://dublincore.org/documents/dcmi-terms/#terms-title) *predicate* is being reused, and the *object* of the triple is the title as listed on the wiki’s index page.

|  |
| --- |
| @prefix dcterms: <http://purl.org/dc/terms/> .  <http://www3.cpdl.org/wiki/index.php/20th\_century>    dcterms:title "20th century" .  <http://www3.cpdl.org/wiki/index.php/21st\_century>    dcterms:title "21st century" . |

NOTE: The use of *triple double quotes* (“””) in the Turtle syntax is legitimate, but [ARC2’](https://github.com/semsol/arc2/wiki)s parser does not handle it properly, so we needed to backtrack and use only *single double quotes* (“).

## 

## D. Apply String Matching Algorithm

<http://en.wikipedia.org/wiki/Levenshtein_distance>  
(X) Survey string matching implementations (off the shelf)

<http://php.net/manual/en/function.levenshtein.php>

Apache StringUtils class: http://commons.apache.org/lang/

(X) Implement joiner (Samuel)

[comparisonModuleJava/Joiner](https://github.com/samuelbjohnson/cross-topix/tree/master/comparisonModuleJava/Joiner)

(X) write [README](https://github.com/samuelbjohnson/cross-topix/blob/master/comparisonModuleJava/Joiner/joinerReadme.txt) that covers usage (How do I run it?)

[comparisonModuleJava](https://github.com/samuelbjohnson/cross-topix/tree/master/comparisonModuleJava) has a jar file

Copy the jar file to the directory where you want to run it

java -jar joiner.jar [firstFileName [secondFileName]]

output is in data.ttl  
  
Java code to create MD5 of concatenation of SPO:

|  |
| --- |
| byte[] bytesOfMessage; try {            // http://stackoverflow.com/questions/415953/generate-md5-hash-in-java            bytesOfMessage   = (myInstanceLocalPrefix + myLocalName).getBytes("UTF-8");            MessageDigest md = MessageDigest.getInstance("MD5");            byte[]     digest   = md.digest(bytesOfMessage);            BigInteger bigInt   = new BigInteger(1,digest);            String     hashtext = bigInt.toString(16);            System.out.println(hashtext + " -> " + hashtext.substring(0,3));  } catch (UnsupportedEncodingException e) {            e.printStackTrace(); } catch (NoSuchAlgorithmException e) {            e.printStackTrace(); } |

Java code to abbreviate the long URIs with prefixes in the Turtle serialization: TODO

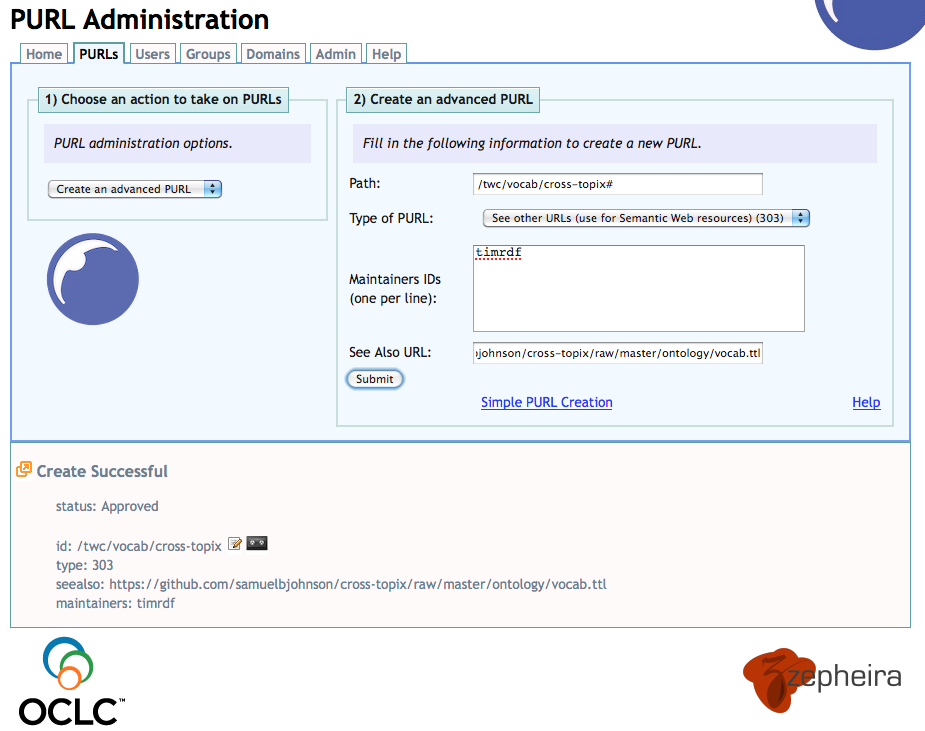
|  |
| --- |
| // conn is your RepositoryConnection from Repository.getConnection(); conn.setNamespace(“comparison”,”http://beta.twc.rpi.edu/id/cross-topix/alpha/”); conn.setNamespace(“xt”,        “http://purl.org/twc/vocab/cross-topix#”); conn.setNamespace(“xsd”,       “http://www.w3.org/2001/XMLSchema#”); |

The namespace for this project’s vocabulary was chosen to align with the [purl.org](http://purl.org/) service, which provides a redirection service when its URIs are requested.

[http://purl.org/twc/vocab/cross-topix#](http://purl.org/twc/vocab/cross-topix) is the namespace for the ontology;

dereferencing it resolves to the file in our github <https://github.com/samuelbjohnson/cross-topix/raw/master/ontology/vocab.ttl>

WWW user interface for the file is available at <https://github.com/samuelbjohnson/cross-topix/blob/master/ontology/vocab.ttl>

  
Figure: This [screenshot](https://github.com/samuelbjohnson/cross-topix/blob/master/ontology/doc/purl-request-and-response.png) documents the creation of the [purl.org](http://purl.org/) vocabulary namespace and its redirection to the [version-controlled OWL document](https://github.com/samuelbjohnson/cross-topix/blob/master/ontology/vocab.ttl) on our [github repository](https://github.com/samuelbjohnson/cross-topix). This enables anyone to dereference URIs in our vocabulary to obtain a formal description for how those terms should be interpretted. It also allows us to control the development of the vocabulary while effortlessly allowing others to access it according to semantic web principles.  
  
() Design output encoding (cite algorithm, give rating for string pairs)

title\_u

title\_p

similarity

heuristic used

|  |
| --- |
| @prefix xsd:        <http://www.w3.org/2001/XMLSchema#> . @prefix dcterms:    <http://purl.org/dc/terms/> . @prefix xt:         <http://purl.org/twc/vocab/cross-topix#> . @prefix comparison: <http://beta.twc.rpi.edu/id/cross-topix/alpha/> .  comparison:551c2c8a0c2a2e07b488d1b8110c116f    xt:comparable\_1 <http://imslp.org/wiki/Ave\_verum\_corpus,\_K.618\_(Mozart,\_Wolfgang\_Amadeus)> ;    xt:comparable\_2 <http://www3.cpdl.org/wiki/index.php/Ave\_verum\_corpus,\_KV\_618\_(Wolfgang\_Amadeus\_Mozart)> ;    xt:similarity "0.36"^^xsd:double  . |

## 

## E. Implementation of the Social Machine:

<http://leo.tw.rpi.edu:81/helloWorld.html>

() Load string matching algorithm results into ARC2 SPARQL endpoint  
() Trial RDF with proposed triples testing with UPDATE and QUERY  
(X) AJAX client to query page similarity results (JQuery)  
(X) display proposed suggestions for user feedback (JQuery)  
() accept and report back human response (Y/N) SPAR/UL

Name of person

Feedback: approve or disapprove suggestion

() learn “enough” SPARQL

<http://logd.tw.rpi.edu/technology/SPARQL> lists some learning resources.

|  |  |  |  |
| --- | --- | --- | --- |
| Source | Dataset | Version | Dataset URI |
| orange | crowd-verifications | 2011-Apr-XX | [2011-Apr-25](http://leo.tw.rpi.edu/source/orange/dataset/crowd-verifications/version/2011-Apr-25) |

*What are the titles of the wiki pages that Amanda hand-selected?* ([results](about:blank))

|  |
| --- |
| prefix dcterms: <http://purl.org/dc/terms/> prefix cpdl: <http://www3.cpdl.org/wiki/index.php/> prefix imslp:   <http://imslp.org/wiki/>  SELECT ?page ?title  WHERE {  GRAPH <http://leo.tw.rpi.edu/source/orange-amanda/dataset/ground-truth/version/2011-Apr-19> {  ?page dcterms:title ?title  } } |

What are the similarities (String Distance) for Amanda’s hand-selected examples? ([results](about:blank)):

|  |
| --- |
| prefix dcterms: <http://purl.org/dc/terms/> prefix cpdl: <http://www3.cpdl.org/wiki/index.php/> prefix imslp:   <http://imslp.org/wiki/> prefix xt:   <http://purl.org/twc/vocab/cross-topix#>  SELECT ?page\_1 ?page\_2 ?sim  WHERE {   GRAPH <http://leo.tw.rpi.edu/source/orange-joiner/dataset/title-similarities/version/2011-Apr-20> {   ?comparison xt:comparable\_1 ?page\_1;               xt:comparable\_2 ?page\_2;               xt:similarity   ?sim;   } } ORDER BY DESC(?sim) |

What are the similarities and titles for Amanda’s hand-selected examples? ([results](about:blank)):

|  |
| --- |
| prefix dcterms: <http://purl.org/dc/terms/> prefix cpdl: <http://www3.cpdl.org/wiki/index.php/> prefix imslp:   <http://imslp.org/wiki/> prefix xt:   <http://purl.org/twc/vocab/cross-topix#>  SELECT ?comparison ?page\_1 ?page\_2 ?sim ?title\_1 ?title\_2  WHERE {  GRAPH <http://leo.tw.rpi.edu/source/orange-joiner/dataset/title-similarities/version/2011-Apr-20> {  ?comparison xt:comparable\_1 ?page\_1;                         xt:comparable\_2 ?page\_2;                         xt:similarity   ?sim .  }  GRAPH <http://leo.tw.rpi.edu/source/orange-amanda/dataset/ground-truth/version/2011-Apr-19> {  ?page\_1 dcterms:title ?title\_1 .  }  GRAPH <http://leo.tw.rpi.edu/source/orange-amanda/dataset/ground-truth/version/2011-Apr-19> {  ?page\_2 dcterms:title ?title\_2 .  } } ORDER BY DESC(?sim) LIMIT 1 OFFSET 0 |

Example insert (use POST):

|  |
| --- |
| prefix dcterms: [<http://purl.org/dc/terms/>](about:blank)  INSERT INTO <http://leo.tw.rpi.edu/source/orange/dataset/crowd-verifications/version/2011-Apr-25> {   <http://hel.lo> dcterms:title "world" . } |

Demonstrate that we actually inserted (use GET) ([results](about:blank)):

|  |
| --- |
| prefix dcterms: <http://purl.org/dc/terms/>  SELECT \* FROM <http://leo.tw.rpi.edu/source/orange/dataset/crowd-verifications/version/2011-Apr-25> {  ?s dcterms:title ?o . } |

This is an example RDF that the Social Machine client will assert back to the SPARQL endpoint after the user provides a “yes” or “no” vote. Demonstrate inserting a similarity vote using real data:

|  |
| --- |
| prefix xsd: <http://www.w3.org/2001/XMLSchema#> prefix xt:  <http://purl.org/twc/vocab/cross-topix#> prefix vote:  <http://leo.tw.rpi.edu/source/orange/dataset/crowd-verifications/version/2011-Apr-25/typed/vote/>  INSERT INTO <http://leo.tw.rpi.edu/source/orange/dataset/crowd-verifications/version/2011-Apr-25> {   vote:myusername\_2011\_04\_25T22\_46\_38\_04\_00  a xt:ComparisonReview;  xt:comparison     <http://beta.twc.rpi.edu/id/cross-topix/alpha/3f041cbbb9eb89a3252bfe0efcd1206d>;  xt:user\_name  "myUserName";  xt:accepted   true;  . } |

What votes have been submited so far?

|  |
| --- |
| prefix xt:   <http://purl.org/twc/vocab/cross-topix#>  SELECT ?comparison ?user ?accepted  WHERE {  GRAPH <http://leo.tw.rpi.edu/source/orange/dataset/crowd-verifications/version/2011-Apr-25> {  ?vote a xt:ComparisonReview;   xt:comparison ?comparison;   xt:user\_name  ?user;   xt:accepted   ?accepted;  .  } } |

SPARQL Query: To update the database to indicate whether the user thought it was a match or not. TODO - Tim.

|  |
| --- |
| # query to put vote onto endpoint |

SPARQL Query: To get the current proposed match with the highest score, TODO

|  |
| --- |
| @prefix  done  ???? |

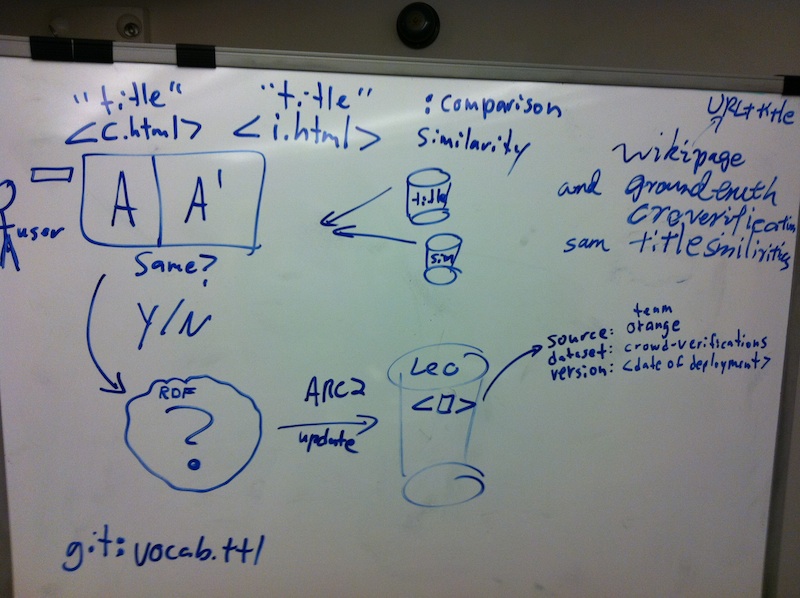


Figure: Enumerating the data attributes to consider when modeling the 1) user response, 2) submitting the response, and 3) choosing the named graph design. This discussion motivates the RDF modeling for the vote (based on the finished RDF modeling of the page titles and similarities), the cross-topix vocabulary (which was extended)

## F. Implementation of the “See Also” box:

<http://leo.tw.rpi.edu:81/hello.php>

(X) Accept URL of [U\_wiki](http://imslp.org/wiki/Stabat_Mater,_Op.58_(Dvo%C5%99%C3%A1k,_Anton%C3%ADn)) or [P\_wiki](http://www3.cpdl.org/wiki/index.php/Requiem,_KV_626_(Wolfgang_Amadeus_Mozart)), the page requested by the user that we are going to provide links to complementing wiki.  
(X) Query “Y/N Aggregation”  
(X) Return suggestions to alternative wiki  
  
We are using csv2rdf4lod’s [c9d.php](https://github.com/timrdf/csv2rdf4lod-automation/blob/master/bin/util/c9d.php) as an example for how to execute SPARQL queries and handle the results. (the node-> part at the top does NOT work b/c we are not in Drupal; we will obtain the input wiki URL using $wiki\_page = $\_GET['page'];, which obtains the attribute from the URL’s HTTP Request:

|  |
| --- |
| <http://leo.tw.rpi.edu:81/hello.php>?page=http%3A%2F%2Fimslp.org%2Fwiki%2FStabat\_Mater%2C\_Op.58\_(Dvo%25C5%2599%25C3%25A1k%2C\_Anton%25C3%25ADn) |

After obtaining the wiki page from the HTTP request, we can construct a SPARQL query to obtain recommendations for links to the other wiki:  
  
See also suggestion box query (“from the left”). The URI highlighted in green is replaced by the page URL requested to hello.php:

|  |
| --- |
| PREFIX dcterms: <http://purl.org/dc/terms/> prefix xt:   <http://purl.org/twc/vocab/cross-topix#>  SELECT ?other ?title ?sim ?user ?accepted  WHERE {  GRAPH <http://leo.tw.rpi.edu/source/orange/dataset/crowd-verifications/version/2011-Apr-25> {  ?vote a xt:ComparisonReview;   xt:comparison ?comparison;   xt:user\_name  ?user;   xt:accepted   ?accepted;  .  }  GRAPH <http://leo.tw.rpi.edu/source/orange-joiner/dataset/title-similarities/version/2011-Apr-20> {  ?comparison xt:comparable\_1 **<http://imslp.org/wiki/Stabat\_Mater,\_Op.58\_(Dvo%C5%99%C3%A1k,\_Anton%C3%ADn)>**;             xt:comparable\_2 ?other;             xt:similarity   ?sim .  }  GRAPH <http://leo.tw.rpi.edu/source/orange-amanda/dataset/ground-truth/version/2011-Apr-19> {  ?other dcterms:title ?title .  } } ORDER BY ?sim ?accepted ?user ?other ?title LIMIT 10 |

See also suggestion box query (“from the right”):

|  |
| --- |
| where { <http://www3.cpdl.org/wiki/index.php/Requiem,\_KV\_626\_(Wolfgang\_Amadeus\_Mozart)> } |

Queries used to “get there”:  
  
What graphs describe or mention a particular page?

|  |
| --- |
| prefix dcterms: <http://purl.org/dc/terms/> prefix cpdl: <http://www3.cpdl.org/wiki/index.php/> prefix imslp:   <http://imslp.org/wiki/> prefix xt:   <http://purl.org/twc/vocab/cross-topix#>  SELECT ?g1 ?s1 ?p1 ?g2 ?p2 ?o  WHERE { GRAPH ?g1 {   ?s1 ?p1 <http://imslp.org/wiki/Stabat\_Mater,\_Op.58\_(Dvo%C5%99%C3%A1k,\_Anton%C3%ADn)> } GRAPH ?g2 {   <http://imslp.org/wiki/Stabat\_Mater,\_Op.58\_(Dvo%C5%99%C3%A1k,\_Anton%C3%ADn)> ?p2 ?o } } |

## 

## G. Install Triple Store and SPARQL Endpoint

(X) Install Triple Store and SPARQL Endpoint (Yu; Thanks [Patrick](http://tw.rpi.edu/instances/PatrickWest)!)

* <https://github.com/semsol/arc2/wiki>
* Human HTML interface is at <http://leo.tw.rpi.edu:81/endpoint.php>
* **http://leo.tw.rpi.edu:81/endpoint.php** is also the Web Service
* NOTE: the endpoint is ONLY accessible from RPI’s network. Use VPN if off campus.
* <http://www4.wiwiss.fu-berlin.de/bizer/rdfapi/tutorial/netapi.html> was NOT used.
* Reference pages:
  + <http://bnode.org/blog/2007/11/26/load-insert-and-delete-in-arc2-via-sparql-plus>
  + <https://help.ubuntu.com/community/ApacheMySQLPHP>
  + <https://github.com/semsol/arc2/wiki>

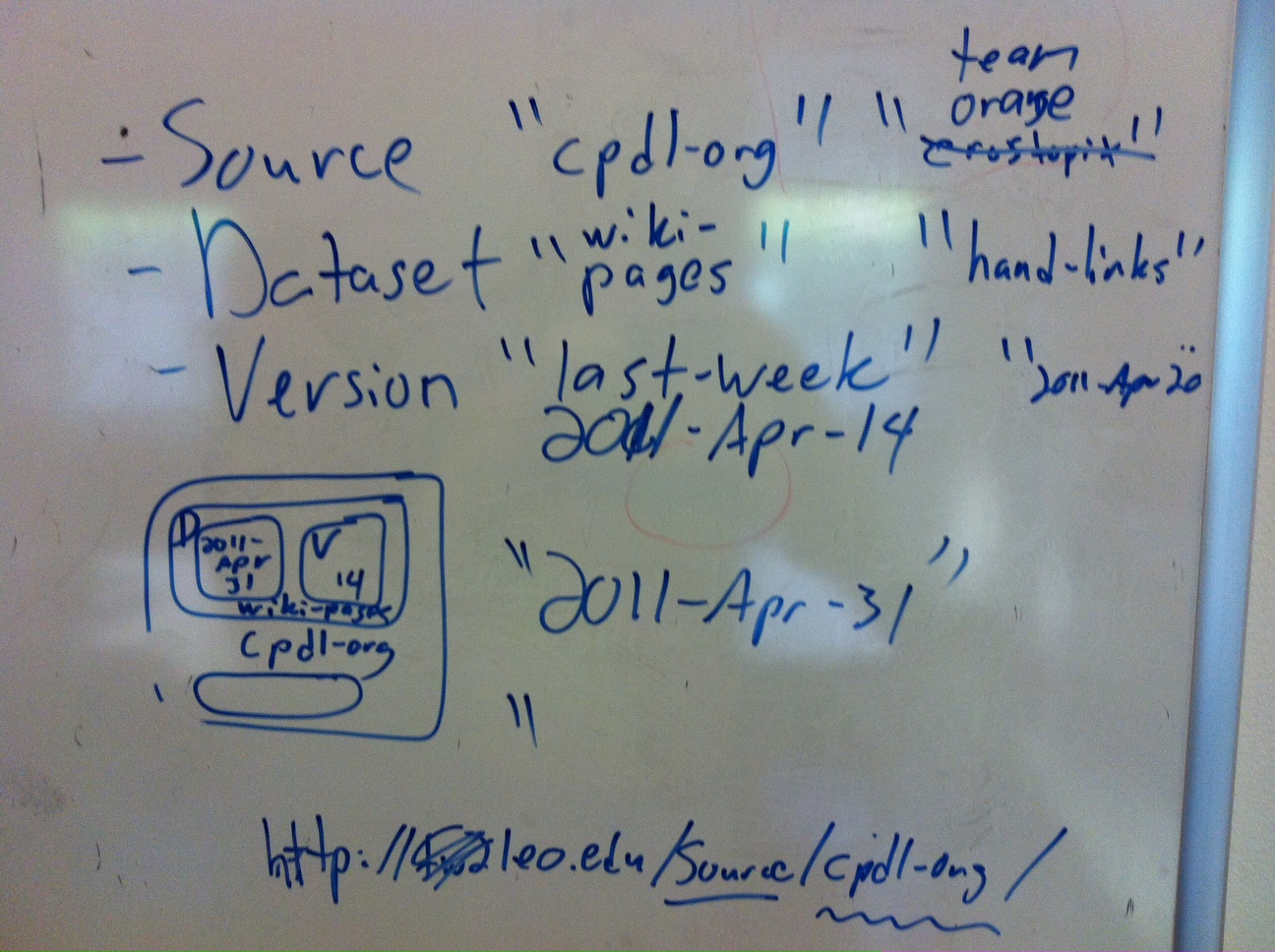


Figure: Illustration of the “three-attribute” contextualizing naming conversion developed by [Lebo, Williams, and Graves](https://github.com/timrdf/csv2rdf4lod-automation/wiki/Publications) in the LOGD project. Providing short identifiers for **source**, **dataset**, and **version** allows for the construction of a dataset’s URI. Our current project has two external sources (**cpdl-org** and **imslp-org**) and several internal sources (**team-orange, orange-amanda, orange-joiner**).

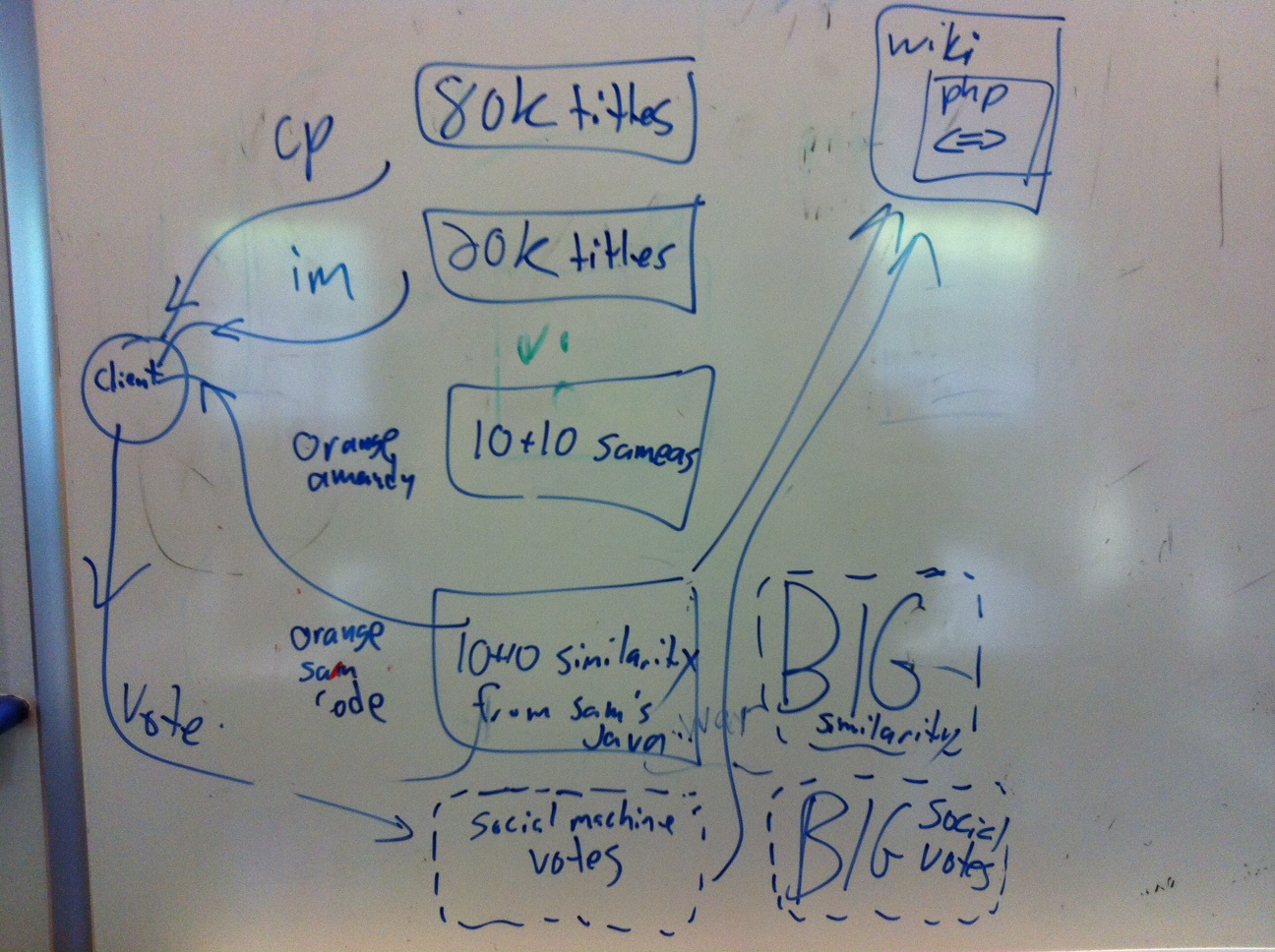


Figure: Whiteboard rendering of the first four named graphs populated in our yet-to-be-installed triple store. “***80k titles - cp***” is the page-titles obtained from scraping CPDL.org, “***20k titles - im***” is the page-titles obtained from scraping IMSLP.org. “***10+10 sameas - orange-amanda***” is a hand-curated list of pages that refer to the common concepts. ”***10+10 similarity from Sam’s Java - orange-samecode***” is the similarity measurements of Amanda’s hand-curated list. The client queries across all three of these to request a confirmation or invalidation from the social machine human, whose vote is placed back into a separate named graph (“***social machine votes”***). The “***wiki php < = >***” produces an HTML widget that suggests links to the second wiki within the wiki pages of the first. The “***BIG similarity***” and “**BIG social votes**” are the larger analogues using the full 80k and 20k page-titles datasets, which we expect to be too large.

() load 80k titles and 20k titles into named graphs  named after their dataset URI  
() load ground truth files into  “  
  
Named graph design strategy:

* Data loaded from TEAM ORANGE will be loaded from our [github repository](https://github.com/samuelbjohnson/cross-topix). The named graph will NOT correspond to the URL from which it was retrieved. Instead, the named graph will be named according to the “Source, Dataset, Version” Dataset URI convention.
* Data loaded from external sources are placed into named graph corresponding to the URL of the original document.  
    
  Load all of the RDF for the **ground truth use case** from the github repository into the [endpoint](http://leo.tw.rpi.edu:81/endpoint.php):

|  |
| --- |
| # NOTE: These are NOT sample queries. These are the queries we use to populate the endpoint.  # # The titles of Amanda’s 10+10 sample use case (14 triples) # DELETE { ?s ?p ?o }  WHERE { GRAPH <http://leo.tw.rpi.edu/source/orange-amanda/dataset/ground-truth/version/2011-Apr-19> { ?s ?p ?o }  } LOAD <https://github.com/samuelbjohnson/cross-topix/raw/master/page-titles/test/hand-picked-amanda.ttl> INTO <http://leo.tw.rpi.edu/source/orange-amanda/dataset/ground-truth/version/2011-Apr-19>  # # Similarities among Amanda’s 10+10 sample use case (300 triples) # DELETE { ?s ?p ?o }  WHERE { GRAPH <http://leo.tw.rpi.edu/source/orange-joiner/dataset/title-similarities/version/2011-Apr-20> { ?s ?p ?o }  } LOAD <https://github.com/samuelbjohnson/cross-topix/raw/master/string-matching\_tests/results\_amanda.ttl>  INTO [<http://leo.tw.rpi.edu/source/orange-joiner/dataset/title-similarities/version/2011-Apr-20>](about:blank)  # # Social machine submitting similarity votes # DELETE { ?s ?p ?o }  WHERE { GRAPH <http://leo.tw.rpi.edu/source/orange/dataset/crowd-verifications/version/2011-Apr-25> { ?s ?p ?o }  } |

Screenshot of ARC2’s Web Browser HTML interface that accepts queries:

|  |
| --- |
| https://lh5.googleusercontent.com/YrTPmwswPNizComhCfhZfejt_Ucn3Q9ZzGF-hWuTp4BwAXYP3JiVlFYDu1HIpuVGwQ_Ga3_n2XBwT75JffJBUzknJ3IyvdaYiMZHQYISCcAAxaVUKVs\* |

*What named graphs are in the endpoint* ([results](about:blank)) (use GET)*?*

|  |
| --- |
| select distinct ?g where {  graph ?g { ?s ?p ?o } } |

Deleting triples in all named graphs (must be POST):

|  |
| --- |
| delete {   ?s ?p ?o }  where {    graph ?g { ?s ?p ?o }  } |

???

|  |
| --- |
| where {} |

???

|  |
| --- |
| where {} |

## H. Future improvements to the system

**1. String Matching Algorithm Improvements**  
To determine if two page titles are similar, we are currently calculating the Levenshtein Distance (also known as the Edit Distance) between the two.  This gives us a number that represents the minimum number of characters that need to be replaced for the two strings to be considered equivalent.  Once we normalized this value to take into the account the length of the strings, we obtained a value that gives us a simple metric to determine whether two titles are similar (and to what degree).    
  
Although this approach works well for a good number of cases, it can be improved to obtain more accurate results.  One way to do this would be to take advantage of the format that these two wikis store their information.  IMSLP.org, for example, stores all their composers with their last name followed by their first name, while CPDL.org does it the other way around.  By calculating the “edit distance” of two composers, we will get a value that is much higher than we would want simply because we are comparing the words out of order.  If there was some way to care the titles on a word-by-word basis rather than as a whole, we can create a more accurate similarity measure.  
  
Another point of improvement is for pages that are for specific composers or genres.  IMPSL.org prefixes all pages of this type with a “Category:” tag which CPDL.org does not do.  Since these extra characters are considered in the calculation of the similarity, our values may be artificially high.  If we ignored “Category:” tags, we might also be able to gather more accurate similarity values.

**2. User scoring evaluation**

Protection for validation of “See Also” box. We would want to prevent malicious users from affecting the results displayed in the “See Also” box.

**3. What next algorithm**

Decides what potential match the user sees in order to continually evaluate the matches and the user selections.

**4. Better Similarity Measures**

Although we were able to get measure better than direct string similarity, there is a wealth of heuristics that could be applied or developed in future iterations. For example, analyzing the narrative in the document itself or use the connectivity among the pages could be used to create different and better measures for similarity.

Multiple measure types leads to handling multiple similarity measures. This would require a modeling change to associate the comparison to a pairing of the quantity with the technique that produced it. We currently have the similarity attribute citing the quantity directly with an implicit understanding that it uses our default similarity heuristic.

**5. Link via Wikipedia**

http://www3.cpdl.org/wiki/index.php/Daniel\_Vetter cites a Wikipedia article

## 

## I. Project Pre-Definition:

**Initial Thoughts**

* This is our group project for <http://tw.rpi.edu/web/Courses/Xinformatics/2011>
* Slide 6 of <http://tw.rpi.edu/media/latest/Xinformatics2011_week6.ppt>
* Analysis of existing information system content and architecture, critique, redesign and prototype redeployment
* Pursuit of a detailed use case around a particular area of informatics includes developing a prototype IS, architecture, design, etc.
* Due May 3 (write up) and May 10 (presentation)
* That’s 7 (8) weeks
* Check in on progress in 3 weeks
* Did Peter outline what we would need to "evaluate", or would we come up with that criteria?
* Evaluate system ideas

**Proposal 1**

* But perhaps "evaluating an existing system" would be easier b/c it is concrete and we can poke it.
* Perhaps we could evaluate the VIVO system. <http://vivoweb.org/>
* It is a professional research "social networking" infrastructure that uses RDF as its representation.

**Use case ideas**

Proposal- Tim ran into some poorly managed MRI handling that we could develop a use case for.

**Meetings**

* Interest in music.
* Amanda is the orchestra librarian.
* Safari books - hard to navigate. Went flash and it is terrible.

**Resources**

* Sheet music - physical stuff. Copyright.
* Libraries
* <http://www3.cpdl.org/wiki/>
* <http://imslp.org/wiki/>
* LOD cloud of music. Music Genome Project (Pandora)
* <http://richard.cyganiak.de/2007/10/lod/lod-datasets_2010-09-22_colored.html>
* [<http://tw.rpi.edu/instances/TimLebo>](about:blank) a foaf:Person .
* <http://tw.rpi.edu/web/person/TimLebo>
* <http://validator.linkeddata.org/vapour>
* <http://dbpedia.org/resource/Composer>  ←----validating dbpedia’s URI for composer:
* <http://validator.linkeddata.org/vapour?vocabUri=http%3A%2F%2Fdbpedia.org%2Fresource%2FComposer&classUri=http%3A%2F%2F&propertyUri=http%3A%2F%2F&instanceUri=http%3A%2F%2F&defaultResponse=dontmind&userAgent=vapour.sourceforge.net>
* <http://www3.cpdl.org/wiki/index.php/Emanuele_d%27Astorga>

2 questions to ask:   
 1) fitting requirements of project  
 2) can we do it in a month.

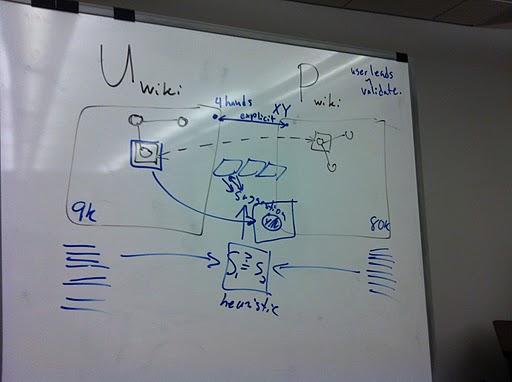
## 

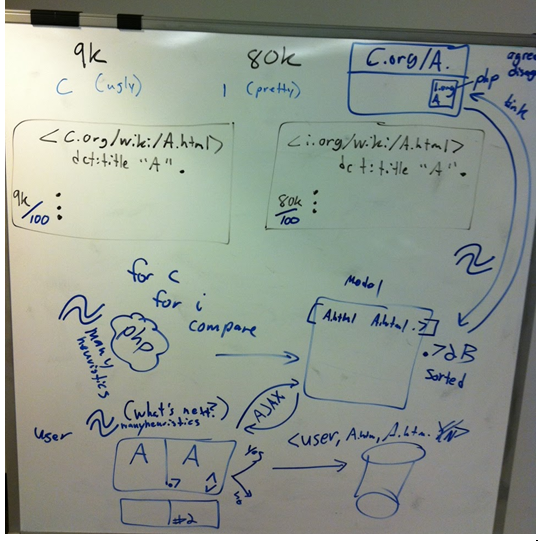
## J. Proposals

Tim defined objectives for proposals  
Done: everyone write 3 paragraphs proposing what we do. (Due Sunday evening) - for review by team on Monday.

* What resources we will use
* What benefits we will provide.
* How we conform to each request in the assignment.
* What technologies we will use
* The kind of user would benefit
* How what we’d do fits into the topics discussed in class
* what deliverable we will provide  
    
  **Proposal by Yu:**  
  We are trying to redesign the digital music library, after analyzing a set of online music libraries, leveraging the social, cognitive and domain concerns that could make a music information system much easier accessible for the music fans. The music libraries that we reference are those such as IMSP-Pertrucci Music Library, Free Choral Sheet Music etc.  
    
  The reason why we redesign the online music library is based on the observation that the pages are information-overloaded and the navigation is neither intuitive nor effective.  These are something we are trying to modify and optimize. In a word, we will redesign the music information system by providing more user-friendly interface that better navigate the user to the piece of information that required. We will also make better arrangement of the presentation of the pages such that the user could immediately get what the pages tells them in a short time.  
    
  As we see the requirements in the assignment, we would definitely draw the diagram of the conceptual model again that better illustrate the infrastructure of all the information system is found in components. What’s more, we will implement a simple prototype of the system within web browser to illustrate our solutions.  
    
  Technologies that we might use are Flash and Action script, PHP etc. All that related to UI design might help. (I independently designed and implemented several Flash and Action scripts apps before, hope it could help in designing our new UI )  
    
  The users that could benefit from our design could be more than the professional musicians. Music lovers without so much domain expertise could also find the scores or recordings.  
    
  To fit into the topics of the materials covered in class, we need definitely consider cognitive, semiotics and social concerns towards a good information system. Therefore, we each might need to be responsible for a particular section of the concerns mentioned in class. In implementing the system, we should collect the suggestions from each of us and realize the functionalities accordingly.  
    
  A deliverables that I could think of is a demo Flash application or a set of linked webpages, which shows how user could better get the information according to our re-newed schemes.  
    
  **Proposal by Sam:**  
  I actually found what Yu wrote to be in line with what my thoughts were, so I’m going to attempt to extend and flesh-out what he wrote, rather than attempt to come up with something completely new (my changes/additions in *italic*):  
    
  **Proposal by Yu (*with extensions/additions by Sam*):**  
  We are trying to redesign the digital music library, after analyzing a set of online music libraries, leveraging the social, cognitive and domain concerns that could make a music information system much easier accessible for the music fans. The music libraries that we reference are those such as IMSP-Pertrucci Music Library, Free Choral Sheet Music etc.  
  *I’d propose focusing on professional or amateur musicians, rather than music fans. The typical music fan isn’t interested in the sheet music itself so much as the product of that sheet music. Because the libraries we discussed on Friday are libraries of sheet music rather than recordings of musical performances, I believe that we should assume any users of our redesigned system would be musicians intending to use the sheet music, rather than fans looking for music to “consume.”*  
    
  The reason why we redesign the online music library is based on the observation that the pages are information-overloaded and the navigation is neither intuitive nor effective.  These are something we are trying to modify and optimize. In a word, we will redesign the music information system by providing more user-friendly interface that better navigate the user to the piece of information that required. We will also make better arrangement of the presentation of the pages such that the user could immediately get what the pages tells them in a short time.  
  *I’m less concerned with the interface itself than the way the information is tagged and organized. We’ll need to come up with a few use cases specifying the reasons a musician comes to one of these sites looking for sheet music, and focus on helping that user achieve his or her goal.*  
    
  *As a semi-professional musician myself, I can both add to our domain knowledge in creating these use cases, as well as provide contacts to professionals in the area who rely on the databases as they currently exist. That will help us when we create the use-cases the project requires.*  
    
  As we see the requirements in the assignment, we would definitely draw the diagram of the conceptual model again that better illustrate the infrastructure of all the information system is found in components. What’s more, we will implement a simple prototype of the system within web browser to illustrate our solutions.  
  *Again, my (and Amanda’s) experience within the domain will be useful in creating a set of diagrams modeling the information. This will be particularly interesting when contrasted with people who have little experience with music--the combination could potentially be very effective.*  
    
  *There are any number of different potential attributes for any given music, and I think that will give a rich complexity to the problem that will make it challenging. While that’s what makes it worthwhile, we should also be aware that it won’t be straightforward. I’m interested to talk more about what Tim started talking about, because I’m not totally sure I understood it. For any given score, there’s an immense amount of available meta-data that could be associated with it, and finding ways to make that meta-data accessible by the users, whether through better searching, or better organization, would be a good direction to go.*  
    
  Technologies that we might use are Flash and Action script, PHP etc. All that related to UI design might help. (I independently designed and implemented several Flash and Action scripts apps before, hope it could help in designing our new UI )  
  *I don’t have any knowledge of Flash or ActionScript (I’m too cheap to pay for Adobe’s development tools); my UI language of choice is JavaScript, but I’m sure we can find a way to sort out that difference.*  
    
  The users that could benefit from our design could be more than the professional musicians. Music lovers without so much domain expertise could also find the scores or recordings.  
  *As I said above, I don’t know that I agree with this, but I’d be interested in hearing more of how a non-musician would make use of musical scores. We need to be clear about whether we’re including recordings within our domain. I believe the only actual sound files that are available on these sites are MIDI versions of the musical scores.*  
    
  To fit into the topics of the materials covered in class, we need definitely consider cognitive, semiotics and social concerns towards a good information system. Therefore, we each might need to be responsible for a particular section of the concerns mentioned in class. In implementing the system, we should collect the suggestions from each of us and realize the functionalities accordingly.  
    
  A deliverables that I could think of is a demo Flash application or a set of linked webpages, which shows how user could better get the information according to our re-newed schemes.  
  *We might also want to consider what Tim talked about i.e., finding ways to integrate these databases into the existing semantic web, which would enable much broader use by the academic community, as well as the application of existing informatics tools and techniques in the searching/browsing of the musical scores. I’d be interested to hear from Tim more about how that could work.*  
    
  **Proposal by Amanda:**  
    
  I agree with most of what has been said above, but there are a few things I would like to add:  
    
  Like Sam had said above, I think the users of this redesigned system will probably be for professional or amateur musicians, although we may not want to only limit ourselves to this.   We can determine the specifics once we create our specific use cases.  
    
  With this in mind, we should have a strong, well-defined search use case planned out since it will be one of the main points of entry to the system.  Along with a search, I think the browse options should also be well-designed.  Because of all the metadata that we could potentially use, browsing can be a very strong feature.  
    
  **Colin’s addition:**  
    
  I agree with much that has been said above so I will add my two cents for what its worth.  
    
  I agree with Sam and Amanda that the functionality of the system should be our main concern. However, I would not disregard the UI as extraneous. Yu’s expertise here could make the system more popular and easier to use for all involved. Before we accept or disregard technologies to use as well, we need to get together and decide what is best for the project.  
    
  I think we could look at this as an educational tool as well. We are focusing on amateur and professional musicians, but we could also include music instructors (choral, band, orchestra) to the mix as well.  
    
  I am concerned with the “Use Case” itself. Reading the instructions for the final assignment, the use case needs to be be well defined and the scope of this project needs to very tight. Based on what I have read, and what I thought the use case should be, we really need to nail that down at Monday’s meeting.  
    
    
  **Proposal by Tim:**
* Steps
  + Install a local Mediawiki
    - Install the Semantic Mediawiki (SMW) extension
    - copy several sample values to local wiki
    - annotate using SMW markup
    - associate annotated pages to LOD cloud and ontologies
    - demonstrate local queries
    - import appropriate ontologies (including FRBR)
    - access the data as RDF dumps.
  + Contact existing wiki owners and convince them to install the sem media wiki extension
    - demonstrate markup
    - train community
    - demonstrate queries
    - ask them what problems they have and document them.
  + explore LOD music
    - establish linking among existing cloud data via the sem media wiki
    - accumulate some of the LOD into a sparql endpoint
    - accumualte some of the wiki RDf into same sparql endpoint
  + crawl existing mediawikis and try to grab some structure
    - encode as RDF establishing URIs as if they had a SMW installation.
  + index physical sheet music
    - don’t publish, just provide metadata and pointers to where it is phsyciall and whom to contact.
  + Additional UI
    - javascript?
    - actioncript?
* Resources:
  + a server machine with admin access to host web stuff
  + SMW software
  + RDF tools/crawlers
  + Music ML?
* Benefits:
  + answering new questions
  + person finding music would benefit
  + linking to Library of Congress?
  + I like Sam’s idea of focusing on musicians proper.
* Deliverables
  + use case documentation
  + prototype system
  + people saying we are awesome and helped them.
  + music ontology

## K. Project Definition and Decomposition



Concept Model v1.2011.04.07

Concept Model v2.2011.04.14

**Everyone get a GitHub account**

* (X) Tim
* (X) Sam
* (X) Yu
* (X) Colin
* (X) Amanda

**Set up version control**

* (X) Start repository (Sam)
* Git-hub link:
* [git://github.com/samuelbjohnson/cross-topix.git](about:blank)
* git@github.com:samuelbjohnson/cross-topix.git

|  |
| --- |
| bash-3.2$ mkdir git bash-3.2$ cd git bash-3.2$ git clone git@github.com:samuelbjohnson/cross-topix.git bash-3.2$ cd into cross-topix/ <edit helloWorld.html - add your name> bash-3.2$ git remote add [repo name] git@github.com:samuelbjohnson/cross-topix.git bash-3.2$ git status bash-3.2$ git diff helloWorld.html bash-3.2$ git add helloWorld.html bash-3.2$ git commit helloWorld.html -m “Some message” bash-3.2$ git push [repo name] master |

\*

## 

## L. Meeting Notes

**A history of meetings Orange Group - Spring 2011**

**Meeting notes: Tuesday 2011 April 05**

Discussed the Possibilities of the Project  
 Instructor: time period, “no half notes”  
 independent musician: I want sheet music. arrangements.  
 create an ontology? keep it small  
 emotional component - music at fourth of july, at a wedding.

**Meeting notes: Thursday 2011 April 07**  
Tonight’s Goals:   
1. Subset of Wiki  
2. Develop example content  
3. How search that?  
4. Small model  
  
 1. We want to Facilitate Browsing  
 2. Be able to address number of instruments  
  
Browsing, NOT search.  
Don’t know composer, style, piece.  
Upbeat music for marching band.  
Given a set of instruments, what works?

**Meeting Notes, Thursday 2011 April 07**

Define Project- much debate and Tim draws nice

**Meeting Notes, Thursday 2011 April 14**

Discussed progress of work  
Set up goals for the night  
 Yu - Make various diagrams for the project  
 Colin and Amanda - Develop use cases  
 Tim - Scraping of two Wiki’s

Samuel – Started the joining of the Wiki – complained at size of file. Almost a throw down over file size

**Meeting Notes, Thursday 2011 April 21**

Discussed progress of work

Yu – created all diagrams for the project, created Sparkle endpoint

Amanda – Created Use cases, ran two cases for similarities

Colin - developed use cases

Tim – Organized the data set management design, reviewed and modified document, triple store installation with Sir Patrick West, coordinated Sparkle with Sir Patrick West and future Doctor Chen (Tim is our master)

Samuel – Finished implementing the joiner, and developed a read me (very lengthy assignment)

Tonight’s work

Yu – Testing triples on Sparkle

Amanda – Coming up with histogram of similarity measure

Colin – Taking notes, documenting work, beginning paper

Tim –

Samuel -

Next steps (after meeting)  
Yu

* vote RDF example - put onto google doc
* ARC2 post and query to verify - put on google doc
* secondary: activity diagram for use case

Amanda

* histogram of 10\*10 DONE
* run on full 1B pages
* string matching future work DONE
* propose threshold and resulting subset size
* starting PHP See Also widget that accepts URL of a wiki page, queries the endpoint for suggestions, and displays it

Samuel

* start social machine UI
* execute SPARQL query

Tim

* design SPARQL query that social machine UI will execute
* design SPARQL query that See Also page query
* verify ARC2 named graph population
* verify Samuel’s similarity output

Colin

* Finish Paper

Semantic MediaWiki: extra syntax give a little semantics:

|  |
| --- |
| ===Publication=== [[composer::Thomas Clark]]'s ''A Third Set of Psalm Tunes with a Magnificat Nunc Dimittis and an Anthem; with an Instrumental Bass'' was published in London by Button & Whitaker of 75, St Paul’s Church Yard. The collection is undated but the ''Hymn Tune Index'' notes that the dates of activity of the publisher at this address were c1808–14, and suggests that this book dates from c1809.  {{#ask:  [[composer:Thomas Clark]] |?title |?date }} |

The following link to the same piece:  
Pretty: <http://imslp.org/wiki/Christ_lag_in_Todesbanden,_BWV_4_(Bach,_Johann_Sebastian)>  
Ugly: <http://www3.cpdl.org/wiki/index.php/Cantata_BWV_4_-_Christ_lag_in_Todesbanden_(Johann_Sebastian_Bach)>  
  
List of potential attributes that we’d want to search on any particular musical score.  
  
<http://data-gov.tw.rpi.edu/wiki/Tim_Lebo>  
  
<http://www3.cpdl.org/wiki/index.php/Daniel_Vetter> cites a Wikipedia article ADD TO FUTURE WORK (Linking via wikipedia)

P.S. Tim has great hair